REMARKS

I. Introduction

In response to the Office Action dated October 4, 2006, claims 1, 6, 7, 12, 13, and 18 have been amended, and claims 19-24 have been added. Claims 1-24 remain in the application. Re-examination and re-consideration of the application, as amended, is requested.

II. Prior Art Rejections

In paragraphs (2)-(3) of the Office Action, claims 1, 3-7, 9-13, and 15-18 were rejected under 35 U.S.C. §103(a) as being unpatentable over Platform SDK: COM IGlobalInterfaceTable (IGLobalInterfaceTable) in view of Tock, U.S. Publication No. 2004/0064570 (Tock). In paragraph (13) of the Office Action, claims 2, 8, and 14 were rejected under 35 U.S.C. §103(a) as being unpatentable over IGlobalInterfaceTable in view of Tock as applied to claims 1, 7, or 13, and further in view of Fujieda, U.S. Patent No. 6,571,203 (Fujieda).

Applicants respectfully traverse these rejections.

Specifically, the independent claims were rejected as follows:

> As to claim 1, IglobalInterfaceTable teaches a computer-implemented method for enabling communication between applications ("…any apartment…any other apartment…" page 1 line 3), comprising: creating a bridge object in a secondary application ("…an object…" page 1 line 1), wherein an interface for the bridge object enables communication with the secondary application through the bridge object ("…an interface…" page 1 line 1); registering the interface for the bridge object in a global interface table (GIT) ("Register…" page 1 lines 5/37-38, "…register…" page 2 line 5); retrieving a cookie from the GIT in response to the registration, wherein the cookie comprises information for utilizing the interface for the bridge object ("…a cookie…" page 2 line 6, "…get a cookie…" page 2 line 5); and storing the cookie in an environment variable, wherein the environment variable is accessible to a application such that the cookie may be retrieved to enable use of the interface ("…GetInterfacefaceFromglobal method…this cookie…" page 1 lines 39-41).
>
> IGlobalInterfaceTable is silent with reference to disconnected applications. Tock teaches disconnected applications ("…offline…" page 1 paragraph 0007, "…disconnected state…" page 9 paragraph 0096).
>
> It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Tock and IGLobalInterfaceTable because the teaching of Tock would improve the system of IGLobalInterfaceTable by providing a method for allowing a client application to operate offline from a server (Tock page 1 paragraph 0007).
>
> As to claims 7 and 13, see the rejection of claim 1 above.

Applicants traverse the above rejections for one or more of the following reasons:

(1) Neither IGlobalInterfaceTable nor Tock teach, disclose or suggest the storage of the cookie in a location that is accessible to a disconnected application;

(2) Neither IGlobalInterfaceTable nor Tock teach, disclose or suggest a disconnected application that is unaware of the secondary application; and

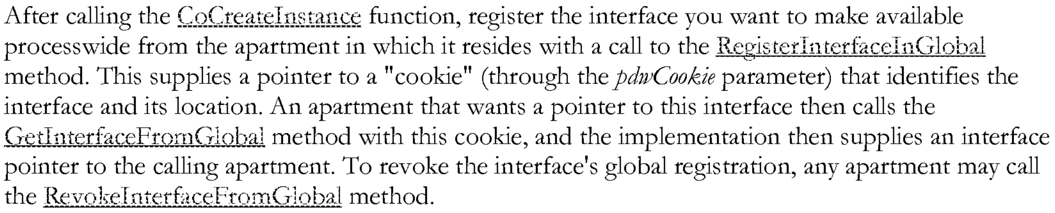
(3) Neither IGlobalInterfaceTable nor Tock teach, disclose or suggest a secondary application and disconnected application executing within the same process but in different apartments.

Independent claims 1, 7, and 13 are generally directed to enabling communication between disconnected applications. More specifically, disconnected applications as used in the present invention and as explicitly claimed therein provide that a disconnected application is unaware of the secondary application. In this regard, the disconnected applications of the present invention are applications that do not know anything about each other. One application (referred to in the claims as a secondary application) creates a bridge object. Such a bridge object is not part of either application and allows the applications to communicate with each other through an interface. In this regard, the claims explicitly provide that an interface for the bridge object enables communication with the secondary application through the bridge object. The interface for the bridge object is registered in a global interface table (GIT) and a cookie is retrieved (from the GIT) in response. Such a cookie comprises information for utilizing the interface for the bridge object. The claims then explicitly provide for storing the cookie in a location that is accessible to the disconnected application such that the cookie can be retrieved to enable use of the interface.

In view of the above-described limitations, there are several unique, novel, and nonobvious aspects of the invention. Such aspects include the storage of the cookie in any globally accessible location. The prior art fails to teach, disclose, or suggest the use or storage of the cookie whatsoever. A second aspect includes that the applications are disconnected/independent and are unaware of each other – yet can communicate via the bridge object. A third aspect is that the interface bridge object is not part of either application. A fourth aspect is that the interface placed in the GIT is from the bridge object, rather than from either application. Again, the applications have no direct connection to each other and are disconnected (yet can communicate via the bridge object and the information contained in the GIT).

The cited references do not teach nor suggest these various elements of Applicants' independent claims.

In rejecting the claims, the Office Action primarily relies on the IGlobalInterfaceTable reference. Specifically, with respect to the storage of the cookie in a location accessible to the disconnected application (e.g., an environment variable), the Office Action refers to page 1, lines 39-41. Page 1, lines 37-41 provides:



> After calling the CoCreateInstance function, register the interface you want to make available processwide from the apartment in which it resides with a call to the RegisterInterfaceInGlobal method. This supplies a pointer to a "cookie" (through the *pdwCookie* parameter) that identifies the interface and its location. An apartment that wants a pointer to this interface then calls the GetInterfaceFromGlobal method with this cookie, and the implementation then supplies an interface pointer to the calling apartment. To revoke the interface's global registration, any apartment may call the RevokeInterfaceFromGlobal method.

As can be seen from this text (and the remainder of IGlobalInterfaceTable), there is no description, explicit or implicit, regarding the storage or what to do with the cookie. Instead, the reference merely describes the supplying of a pointer to a cookie that identifies an interface and its location. A method is then called with the cookie and an interface is supplied to a calling apartment. In this regard, the cookie of IGlobalInterfaceTable could merely be passing around the cookie as an argument to various functions. The present claims are unique in that the cookie of the claimed invention is stored in a location that is accessible to both the disconnected application and the secondary application. For example, the cookie could be stored in a globally accessible location such as a database, file system, or registry. The dependent claims explicitly provide that the location comprises an environment variable. Nowhere in IGlobalInterfaceTable is there any description, suggestion, or remote reference to the storing of the cookie in any location, not to mention the storage in an environment variable as claimed.

The Office Action continues and submits that IGlobalInterfaceTable is silent with respect to disconnected applications. Instead, the Office Action relies on Tock to refer to a client and server. Applicants respectfully traverse such an assertion. Namely, the disconnected applications of the present invention do not refer to disconnected in the sense of a remote server that is "disconnected" from a client application (e.g., as in Tock). Instead, the amended claims provide that the disconnected application is unaware of the secondary application. Such use of the term "disconnected" is not even remotely similar to that of client/server disconnection described in Tock and asserted in the Office Action. Further, the dependent claims provide that the applications are executing within a same process but in different apartments. Such terminology is wholly and completely lacking from Tock.

The remaining cited references fail to cure the deficiencies of IGlobalInterfaceTable and Tock.

Moreover, the various elements of Applicants' claimed invention together provide operational advantages over IGlobalInterfaceTable, Tock, and Fujieda. In addition, Applicants' invention solves problems not recognized by IGlobalInterfaceTable, Tock, and Fujieda.

Thus, Applicants submit that independent claims 1, 7, and 13 are allowable over IGlobalInterfaceTable, Tock, and Fujieda. Further, dependent claims 2-6, 8-12, and 14-18 are submitted to be allowable over IGlobalInterfaceTable, Tock, and Fujieda in the same manner, because they are dependent on independent claims 1, 7, and 13, respectively, and thus contain all the limitations of the independent claims. In addition, dependent claims 2-6, 8-12, and 14-18 recite additional novel elements not shown by IGlobalInterfaceTable, Tock, and Fujieda.

III. Conclusion

In view of the above, it is submitted that this application is now in good order for allowance and such allowance is respectfully solicited. Should the Examiner believe minor matters still remain that can be resolved in a telephone interview, the Examiner is urged to call Applicants' undersigned attorney.

Respectfully submitted,

GATES & COOPER LLP
Attorneys for Applicant(s)

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: January 2, 2007

By:/Jason S. Feldmar/
Name: Jason S. Feldmar
Reg. No.: 39,187

JSF/